Simple email encryption with FireGPG

Simply put, cryptography[1] is the process of hiding information. Though long the domain of computer scientists, mathematicians and secretive government agencies, this technology is now both fairly easy to use and ubiquitous. This spread of encryption technology has not happened a moment too soon, as encryption[2] is more important for the average person than ever before.

Every email you send or receive, every instant message, comes to and from your computer without any protection. Due to the decentralized nature of the Internet, all these personal communications pass through dozens of computers, in some cases spread out through several countries, in the milliseconds it takes for message to go from sender to receiver. Many of these computers, particularly your email provider, are required by law to keep copies[3] of all of your messages, for long periods of time. Your private[4]online conversations are not as private as you thought.

When faced with the reality that their online communication are not very private, many people claim that they "have nothing to hide." According to Professor Solve[5] of George Washington University Law School, "the problem with the nothing to hide argument is with it's underlying assumption that privacy is about hiding bad things." Have you ever had a very personal conversation with a close friend? Did you tape record that conversation and email it to everyone on your email address book? If not, then you probably have something to hide. It is not that such things are bad, it is that they are private.

Also, many people in the world live in countries that are not friendly to the idea of free speech[6], countries that suppress political and religious ideas that go against the government sanctioned norm. Many people in this situation try to get around government censorship and espionage by misspelling certain sensitive words or replacing them with "code words." Though such substitution is of limited usefulness in the case of automated censorship, it is of little use against actual espionage.

For all the various email privacy needs, there is a simple and easy to use solution known as the GNU Privacy Guard (GPG)[7]. Though the type of cryptography used by GPG is so secure that many governments use it to secure top secret information, GPG by itself is not very user friendly. However, FireGPG[8], an extension for Firefox[9], is an easy to use interface for GPG that can be used right inside your web browser. This guide focuses on installing and using FireGPG to send and receive encrypted email messages, on Windows, OSX and Linux.

Prerequisites

GnuPG is very secure. However, the best security can be undermined by users who make mistakes, and don't think their actions through clearly. It is not uncommon for people to use email encryption, only to have access of their private key stolen because they have a weak password. Also, if your computer has already been compromised by an attacker (perhaps by a virus or other means), then it is trivial for that attacker to steal your private key and your passwords, making it easy to intercept all your private communications. For a simple guide on staying safe on the Internet, check out this post[10].

The target audience for this guide is the average computer user. If you know how to browse the web and send email, this guide is for you. You do not need to be a computer geek, but you do need to be willing to think. Nothing in this guide should be prohibitively difficult to understand, but you will need to learn a few new concepts. If you are willing to keep an open mind, then read on.

Two types of encryption

Symmetric encryption

Imagine you are trying to send a private message to a group of friend, but you do not trust the mailman. The solution is to find a way to hide the message in plain sight. To do this you need three things: a message to send, a secret shared with your friends, and an agreed upon process for using the shared secret to hide the message in what looks like gibberish.

This is what is known of as *symmetric key cryptography*[1], because there is only one key, aka secret, involved. The process for combining the key with the message is known as a cryptographic algorithm ("algorithm" is simply a fancy way of saying "set of instructions"), sometimes called a cypher[11]. Since cryptography is much easier to use than it once was, a user does not need to concern themselves with the intricate mathematical details of how the algorithm works.

Asymmetric encryption, aka public key encryption

There is one simple problem with symmetric encryption: it does not scale well. This is not so much a problem with the technology as it is a problem with the people using the technology. Everyone knows that a secret shared with two people is twice as difficult to keep secret as the same secret kept by only one person. Shared with three people it is three times as difficult, and with four people... you get the idea. There is also the issue of how to safely share the secret with groups of people. Again the more people you add, the more difficult it becomes.

The answer to this problem is what is known as *public key cryptography*[1]. With this kind of cryptography, the analogy of a key works less well than it did with symmetric encryption. Imagine that you again want to share a secret message with your group of friends. However this time, the cryptographic algorithm you have agreed to use is such that each person in the group has two keys. You do not need to understand all the fancy mathematical details, but what you do need to understand is that unlike one key cryptography, *each key in your pair can encrypt a message that the other key, and only the other key, can decrypt*. This time everyone takes one of those keys and publishes them, letting the whole world see it. This is what is known as the public key. The second key is carefully hidden and protected, and is known as the private key.

It works like this: Since you published your public key, any of your friends can find it on the Internet and use it to encrypt a secret message to you. Once it is encrypted with your public key, you are the only person who can decrypt it, because you keep your private key secret. This solves the

problem of having a large group share a single secret.

This also lets you do something else, and that is prove your identity. As I said earlier, if something is encrypted by one key, the other key can decrypt it. Since you keep your private key secret, you can encrypt messages with it that only your public key can decrypt. Assuming that everyone assumes that you keep your private key safe, then anyone can decrypt that message you encrypted with your private key, while you are the only person who could have possibly encrypted it, therefore proving your identity. This is what is known as a signature. It is common practice for a sender to encrypt, or "sign" messages with their private key, then encrypt the message again with the public key of the recipient. When the recipient decrypts the message first with their private key and then with the sender's public key, they know that not only are they the only person to have read the message, but that the message was sent by the person who said they sent it.

Public key cryptography is the type of encryption that GPG uses. The key pair mentioned above is actually a set of two files containing very large mathematically related random numbers. The rest of this guide will walk you through installing GPG on your computer, generating a set of keys, sharing public keys and using them to send encrypted messages. Make sure you understand the concept of public key cryptography before reading on.

Using encryption

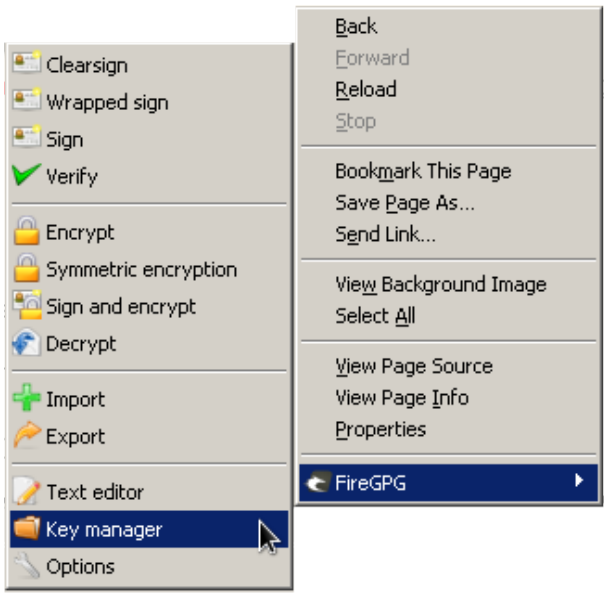
Installing GnuPG and FireGPG

To start off with, you need to have GPG installed on your system. If you are running Linux, then chances are it is installed by default. If you are running Windows or OSX, you need to download the Windows installer[12] (gnupg-w32cli-1.4.9.exe) or the OSX installer[13] (MacGPG2-2.0.10-2.zip).

Once GPG is installed, open up Firefox and head over to getfiregpg.org[8]. Click on the `install` link to the right and click `Download FireGPG` on the next page. At this point you should see a warning at the top of the page saying "Firefox prevented this site (getfiregpg.org) from asking you to install software on your computer." Click `Allow`, and then `Install Now` on the window that pops up. In a few seconds Firefox will tell you that it needs to restart the browser for the installation to take effect, so go ahead and do that.

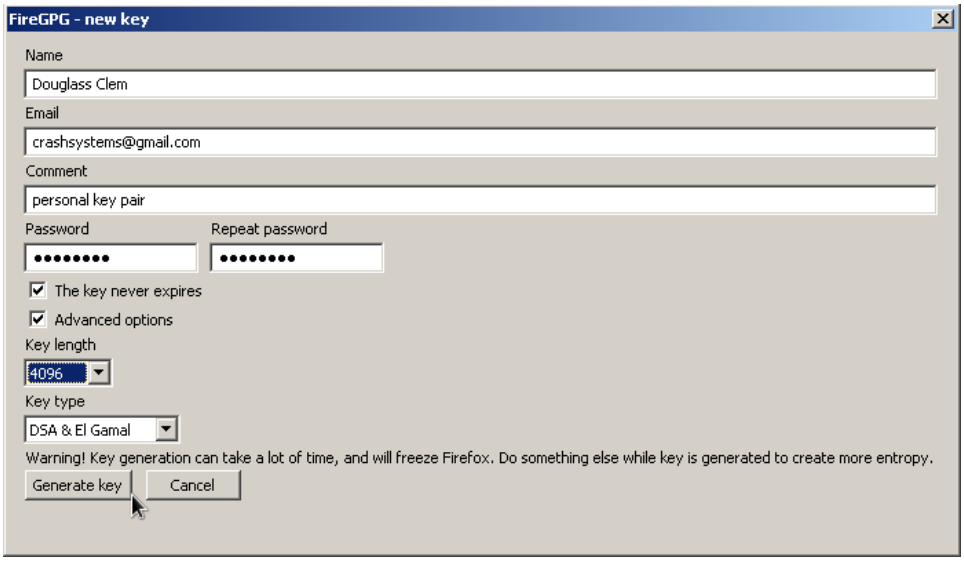
Generating a key pair

The first thing you need to do once you have GPG and FireGPG installed is generate your personal key pair. You can access all of FireGPG's features via the right-click menu. To open up the key manager, right-click somewhere in the page and go to `FireGPG/Key manager`.



It is important to note that when you make a new key, you need a *really* good password[10]. Hopefully your private key never falls into the wrong hands, but if it does it will be this password that protects it. In fact, the password is used as the key to a form of symmetric encryption that protects the private key you generate. If you have a good enough password, then the only practical way for an adversary to gain access to your key is to force you to reveal your password (interrogation, torture, blackmail, etc.).

At the bottom of the key manager window, click `New Key`. Fill out the information on the form, making sure to check boxes `The key never expires` and `Advanced options`. Under `Key length`, set the value to 4096. Click `Generate key` and take notices of the text above the button, which reads "Warning! Key generation can take a lot of time, and will freeze Firefox. Do something else while key is generated to create more entropy." Firefox froze on my system for about 10 minutes while the key was being generated. This is normal, though your key generation time will vary based upon the speed of your processor. What the second bit of the warning text means is that you should move your mouse around a lot while the key is being generated. This will help make the process go a little faster.

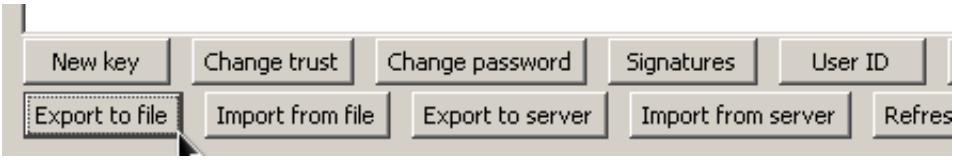


Backing up your keys

The first thing you need to do once your key pair is generated is to back them up. There is nothing worse than distributing your public key to all your friends and coworkers, only to have your laptop lost or stolen, or have a hard drive break. These backups should generally be off-line, so as not to make it easier for an adversary to gain access to your keys remotely. If you have a CD burner, CDs make for great backups. Burn the files to CD and then stick it in a fire proof safe. If you do not have such a safe, give a copy to a trusted friend for safe keeping. If a CD is not an option, than a cheap USB stick will work as well. On OSX and Linux your keys will be in the folder `.gnupg` inside your home folder, so copy the entire folder (you may need to reveal hidden files first). On Windows, your gnupg folder is located at `C:\Documents and Settings\admin\Application Data\gnupg`, where `admin` is your user name.

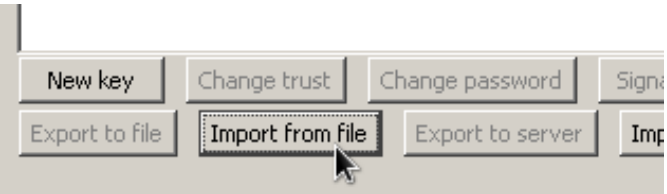
Sharing a key

Once you have created your keys and have backed them up, you need to start giving people your public key, so they can send you private messages. Make sure your key is selected in the key manager, then click `Export to file`. Give it a good name, like `myname.asc`, then save it to your desktop. You can now put this on a USB stick to give to a friend, email it as an attachment, or any other method of sending a file to a friend.



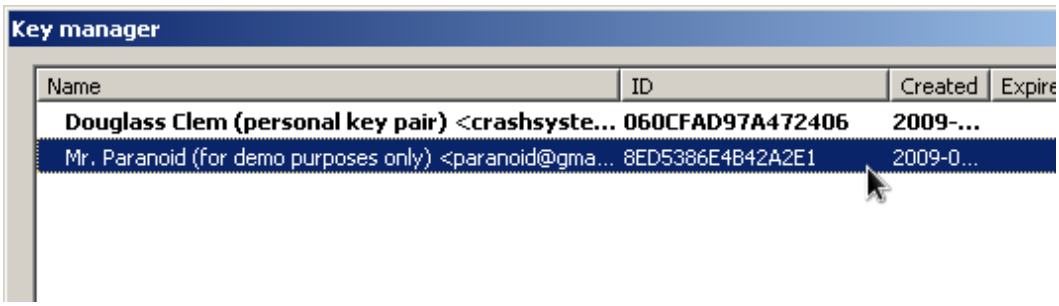
Importing and validating a public key

Now lets say that Mr. Paranoid just exchanged public keys with you. To be able to encrypt messages with his public key you must first import it. To do so, go into the key manager and click `Import from file`. Find the key file in your file manager and click `open`.



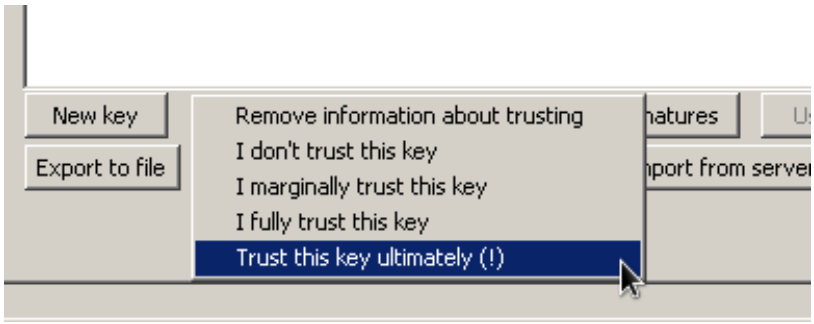
When you import a person's key, it is very important to determine your level of trust of that key, and there are a number of factors that go into this decision. First, you need to decide if you actually trust this person (in the normal sense of the word). Second, it is helpful to know if the person safely manages his keys. Ask him if he has a strong password, and discuss what the meaning of a strong password[10] is. Secondly, if he is running Windows, does he have anti-virus software installed, and does he keep it up-to-date? Does this person run regular system updates and update the software on his machine? It is dangerous to send private message to someone who is lazy about protecting their private key.

Also, perhaps the most important part about establishing trust is verifying that the public key you have came from who it says it came from. If you do not do this, then anyone can email you a key saying they are someone you know, even if they are not. This is what is known as a man-in-the-middle attack. The best way to establish this part of trust is to exchange the keys in person, face-to-face. Every key has a unique ID, which you can view in the key manager. Once the keys are exchanged, read off one another's key IDs to verify that you have the right key.



In some cases it is simply impossible to verify keys in person. When this is the case, video chat (using a program such as Ekgia[14]or Skype[15]) is a good second option. Start a video conference, and read off the key IDs to one another just like you would if you were in person. Once you have determined the level of trust you are comfortable with, ether in person or via video chat, you need to

assign this to the public key you imported. Select the key in the key manager, then click `Change trust`. You will see a menu listing various statements describing various levels of trust, so click the statement you are most comfortable with.

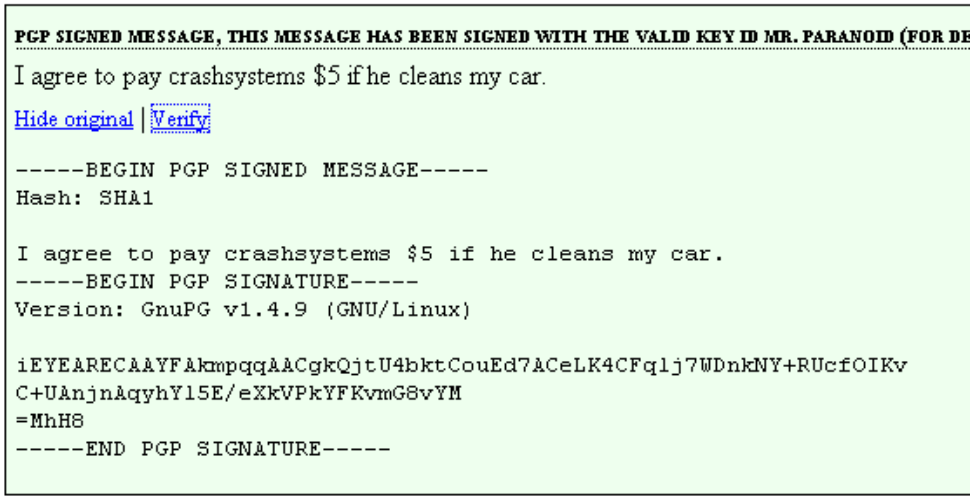


Signing & verifying text

Now that we have some keys exchanged, it is time to learn about signatures and encryption. Sometimes you may want to write text that anyone can read, but want a way to prove that you wrote it. Right-click on the page, go to the FireGPG menu, and select `Text editor`. Type the message that you want to sign, then click `Clearsign`. You'll be presented with a list of your private keys (which will only have one item if you've only generated one key pair). Click your private key and click `Ok`. Type your password when prompted, then press enter. You will now have the signed text in the text editor, so you can click `Copy to clipboard and close` and then paste it in an email, web page or wherever you want to place signed text.

Verifying signatures with FireGPG is very easy. By default FireGPG will detect blocks of text in a web page (or web-based email account) that are signed. It will also hide the signature data by default, and only display the signed text. Click `Verify` to see if the signature can be validated by a public key you have in your collection. Also, if you want to see the full signed message, click

`Display original.`



Encrypting text with a public key

When you are writing an email in a web-based email service, and you plan on encrypting that email, it is important that you use FireGPG's text editor. This is because most web mail providers have an auto save feature, which saves your draft email to their server every few minutes. This means that the email provider has a copy of your unencrypted email. Once you are done typing your email in the text editor, click `Encrypt`. First you will be asked to select the public key(s) to encrypt to (you can select more than one by holding down the 'ctrl' key while clicking). When you click `Ok`, you will next be asked for the private key you want to sign the message with. If you do not want to sign the message, click cancel. If you do choose to sign the message, you will be asked for your private key's password. Once the message is encrypted, you can click `Copy to clipboard and close`, and then paste it into the email you are going to send.

Decrypting text

Blocks of encrypted text in web pages are detected by FireGPG in exactly the way it detects signed text. Click `Decrypt`, and then type your password when prompted. Once you hit enter, you will see the original message, plus an indication if the message was properly signed by the sender or not (if the sender signed the message). You can also click `Display original` to see the encrypted text.

References

1. *Cryptography (Wikipedia)*: http://en.wikipedia.org/wiki/Cryptography
2. *Encryption (Wikipedia)*: http://en.wikipedia.org/wiki/Encryption
3. *EFF.org Search Results, "data retention"*: http://www.eff.org/search?text=data+retention
4. *Privacy (EFF)*: http://www.eff.org/issues/privacy
5. *'I've Got Nothing to Hide' and Other Misunderstandings of Privacy*: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=998565
6. *Free Speech (EFF)*: http://www.eff.org/issues/free-speech
7. *GNU Privacy Guard*: http://gnupg.org/
8. *FireGPG*: http://getfiregpg.org
9. *Mozilla Firefox*: http://mozilla.org
10. *The Comprehensive Guide To Safe Web Browsing*: http://crashsystems.net/2008/10/safe-web-browsing/
11. *Cipher (Wikipedia)*: http://en.wikipedia.org/wiki/Cipher
12. *Index of ftp://mirror.cict.fr/gnupg/binary/*: ftp://mirror.cict.fr/gnupg/binary/
13. *SourceForge.net, Mac GNU Privacy Guard v2.x Files*: http://sourceforge.net/project/showfiles.php?group_id=248469&package_id=303406
14. *Ekiga ~ Free your speech*: http://ekiga.org/
15. *Skype official website*: http://skype.com/

Copyright